Available online at www.sciencedirect.com
SCIENCE DIRECT®
Discrete Applied Mathematics 148 (2005) 107–125
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

Conversion of coloring algorithms into maximum weight independent set algorithms☆

Thomas Erlebach[a,*], Klaus Jansen[b]

[a] *Department of Computer Science, University of Leicester, Leicester LE1 7RH, UK*
[b] *Technical Faculty of Christian-Albrechts-University of Kiel, Institute of Computer Science and Applied Mathematics, Olshausenstr. 40, D-24098 Kiel, Germany*

Received 10 July 2003; received in revised form 27 July 2004; accepted 16 November 2004

Abstract

A general technique for converting approximation algorithms for the vertex coloring problem in a class of graphs into approximation algorithms for the maximum weight independent set problem (MWIS) in the same class of graphs is presented. The technique consists of solving an LP-relaxation of the MWIS problem with certain clique inequalities, constructing an instance of the vertex coloring problem from the LP solution, applying the coloring algorithm to this instance, and selecting the best resulting color class as the MWIS solution. The approximation ratio obtained is the product of the approximation ratio with which the LP formulation can be solved (usually equal to one) and the approximation ratio of the coloring algorithm with respect to the size of the largest relevant clique. Applying this technique, the best known approximation algorithms are obtained for the maximum weight edge-disjoint paths problem in bidirected trees and in bidirected two-dimensional meshes with row–column routing, and for time-constrained scheduling of weighted packets in the same topologies. These problems are also proved to be MAX SNP-hard.

Keywords: Approximation algorithms; Weighted independent set; Edge-disjoint paths; Path coloring; Time-constrained packet scheduling; Linear programming

☆ A preliminary version of this paper was presented at the Workshop on Approximation and Randomization Algorithms in Communication Networks ARACNE 2000, Geneva, July 15, 2000.

* Corresponding author.
E-mail addresses: t.erlebach@mcs.le.ac.uk (T. Erlebach), kj@informatik.uni-kiel.de (K. Jansen).

0166-218X/$ - see front matter
doi:10.1016/j.dam.2004.11.007

1. Introduction

The maximum independent set problem (MIS) is a fundamental problem in graph theory with many applications. Given a graph $G = (V, E)$, the goal is to compute a subset of the vertices such that no two vertices in the subset are joined by an edge (such a subset is called an independent set) and such that the cardinality of the subset is maximized. In most applications, G represents a conflict graph (an edge joining two vertices u and v indicates that at most one of u and v can be selected) and its vertices represent desirable objects; the goal is to select as many of the desirable objects as possible while not selecting two conflicting objects.

The maximum weight independent set problem (MWIS) is an important generalization of MIS: the vertices of the given graph G are assigned weights, and the goal is to compute an independent set I in G such that the sum of the weights of the vertices in I is maximized. The weights of the vertices can be thought of as measuring the importance or the benefit associated with the corresponding objects.

Examples for applications of MIS and MWIS are disjoint paths problems in networks [9], interval selection problems arising in manufacturing [31], map labelling problems [35], and frequency assignment problems [29].

Bar-Noy et al. [5] obtained an approximation algorithm for throughput maximization on multiple machines, a special case of MWIS, using an LP-relaxation and a rounding procedure based on a coloring subroutine. In this paper, we extend their approach and present a general technique for converting vertex coloring algorithms for a class of graphs into MWIS algorithms. The technique consists of solving an LP-relaxation of the MWIS problem (where certain clique inequalities are added to the LP formulation), constructing an instance of the vertex coloring problem from the LP solution, applying the coloring algorithm to this instance, and selecting the best resulting color class as the MWIS solution. The approximation ratio obtained is the product of the approximation ratio with which the LP formulation can be solved (equal to 1 if the LP can be solved optimally in polynomial time) and the approximation ratio of the coloring algorithm with respect to the size of the largest relevant clique. Applying this technique, we obtain new and best known approximation algorithms for the maximum weight edge-disjoint paths problem (MWEDP) in bidirected trees and in bidirected two-dimensional meshes with row–column routing, and substantially improved approximation algorithms for time-constrained scheduling of weighted packets (TCSWP) in the same topologies. The resulting approximation algorithms achieve an approximation ratio for the weighted case that matches the approximation ratio of the best known algorithms for the unweighted case and is substantially better than previously known algorithms. We expect that the technique will find many further applications (e.g., interval scheduling, frequency assignment, geometric graphs).

While most of the ingredients of the technique were already present in the work of Bar-Noy et al. [5], our contribution lies in formulating and analyzing the technique in a general setting, and in demonstrating its applicability by using it to derive best known approximation algorithms for several other problems.

For the throughput maximization problem studied in [5], combinatorial algorithms that achieve the same approximation ratio as the LP-based algorithms of [5] have recently been found by Bar-Noy et al. [4] and by Berman and DasGupta [6]. We can show that their techniques can also be applied to the problems we study in Sections 4 and 5, giving approximation algorithms with ratio 2 for MWEDP in trees, 4 for MWEDP in meshes with row–column routing, 3 for TCSWP in trees, and 6 for TCSWP in meshes. Except in the case of MWEDP in trees, these ratios match the ratios that we obtain in Sections 4 and 5 with our LP-based technique. Nevertheless, we believe that the LP-based technique remains useful for the following reasons:

(1) The LP-based technique is very general and can use an arbitrary coloring algorithm as a subroutine. The combinatorial algorithms of [4,6] appear to give the same approximation ratio as the LP-based technique only in those cases where a good coloring can be obtained by a simple greedy algorithm.
(2) The LP-based technique appears to produce much better solutions in practice than indicated by the worst-case bounds that we derive. For example, experimental results for MWEDP in trees indicate that the solutions obtained with the LP-based technique are better than those obtained with combinatorial algorithms on randomly generated instances [10].

1.1. Related work on independent sets

MIS has long been known to be $\mathcal{NP}$-hard for general graphs [13]. Even in graphs whose maximum degree is bounded by 3, MIS is MAX SNP-complete [7]. More recently, MIS was shown to be very hard to approximate in general: unless $\mathcal{NP} = \text{co-}RP$, there can be no polynomial-time approximation algorithm with ratio $n^{1-\varepsilon}$ for any $\varepsilon > 0$ in graphs with n vertices [21]. (Note that an independent set in G is a clique in the complement of G.) For restricted classes of graphs, the situation is different. For example, MIS and MWIS can be solved optimally in polynomial time in perfect graphs [17] (e.g., interval graphs, permutation graphs, transitive graphs [22]), in partial k-trees with constant k [3], in circular-arc graphs, or in the edge-intersection graphs of undirected paths in a tree (if the path representation is given) [33]. For other restricted classes of graphs, MIS and MWIS are $\mathcal{NP}$-hard, but can be approximated in polynomial time within a constant factor: for example, a $(\frac{5}{3} + \varepsilon)$-approximation for MIS is possible for the edge-intersection graphs of directed paths in bidirected trees [9]. A survey of known approximation results for MIS and MWIS in general graphs and in graphs of bounded degree is given in [19]. Improved approximation results for MWIS are reported in [20].

1.2. Related work on coloring and fractional coloring

Grötschel et al. [16] have shown that the weighted fractional coloring problem is $\mathcal{NP}$-hard for general graphs, but can be solved in polynomial time on perfect graphs. Specifically, they have proved the following result: for any graph class $\mathcal{G}$, if the problem of computing the size of the largest independent set in G, for graphs $G \in \mathcal{G}$, is $\mathcal{NP}$-hard, then the problem of determining the weighted fractional chromatic number is also $\mathcal{NP}$-hard. This gives a negative result for the weighted fractional coloring problem even for planar cubic graphs.

Furthermore, if MWIS is polynomial-time solvable for graphs in $\mathscr{G}$, then the weighted fractional coloring problem for $\mathscr{G}$ can also be solved in polynomial time.

Lund and Yannakakis [28] proved that there is a $\delta > 0$ such that there does not exist a polynomial-time approximation algorithm for fractional coloring that achieves ratio $|V|^{\delta}$, unless $P = \mathscr{N}\mathscr{P}$.

Feige and Kilian [12] proved that the chromatic number cannot be approximated within $|V|^{1-\varepsilon}$ for any $\varepsilon > 0$, unless $\mathscr{N}\mathscr{P} \subseteq$ ZPP. This also holds for the fractional coloring problem, due to the logarithmic relationship (see also [27]) of the fractional chromatic number and the chromatic number (i.e., $\chi_{\mathrm{f}}(G) \leqslant \chi(G) \leqslant \chi_{\mathrm{f}}(G)(1 + \ln s(G))$, where $s(G)$ is the size of the largest independent set in G).

1.3. Summary of results

In Section 2, we give formal definitions of the concepts used in the paper and introduce the classes of graphs that we call (α, β)-tractable, (α, β)-fractional-tractable, and (α, β)-weighted-fractional-tractable. Roughly speaking, these are classes of graphs that admit solving a linear programming relaxation of MWIS within a factor of β of the optimal solution and computing colorings (resp. fractional colorings and weighted fractional colorings) within a factor of α of the largest relevant clique.

In Section 3, we present our main result, a general technique for converting coloring algorithms into MWIS algorithms. We show that for every (α, β)-tractable or (α, β)-fractional-tractable class of graphs, there is an approximation algorithm for MWIS with ratio $\alpha\beta + \varepsilon$, where ε is an arbitrarily small positive constant. For every (α, β)-weighted-fractional-tractable class of graphs, there is an approximation algorithm for MWIS that achieves approximation ratio $\alpha\beta$.

Then we turn to applications. In Section 4, we study the MWEDP, where the input consists of a bidirected graph and a set of directed paths with arbitrary weights and the goal is to select a subset of edge-disjoint paths of maximum total weight. Our technique gives a $(\frac{5}{3} + \varepsilon)$-approximation for MWEDP in trees (matching the result in [9] for the unweighted version of the problem) and a 4-approximation for MWEDP in meshes with row–column routing. For meshes with row–column routing, we also establish that MEDP is MAX SNP-hard and path coloring is $\mathscr{N}\mathscr{P}$-hard; the proofs of these two results are deferred to Appendix A and Appendix B, respectively.

In Section 5, we study the problem of time-constrained packet scheduling. Given a set of weighted packets with release times and deadlines, the goal is to schedule the packets such that the total weight of packets delivered by their deadlines is maximized. For the bufferless case, we obtain a 3-approximation for tree networks and a 6-approximation for meshes with row–column routing, substantially improving on previously known algorithms from [1].

2. Preliminaries

Let $G = (V, E)$ be an undirected graph. We let $\omega(G)$ denote the cardinality of the largest clique in G. By $\omega(G, c)$ we denote the total weight of a heaviest clique in G with respect to a weight vector c, i.e., the total weight of a clique C that maximizes $\sum_{v \in C} c_v$.

We will consider classes of graphs together with a function $\mathscr{C}$ that maps every graph G from the class to a set of cliques in G. We then call $\mathscr{C}(G)$ the *relevant cliques* of graph G. For example, $\mathscr{C}(G)$ could be the set of all maximal cliques in G. Or, if G is a circular-arc graph, $\mathscr{C}(G)$ could be the set of all cliques of arcs overlapping one point of the circle. For a given function $\mathscr{C}$, we denote by $\omega_{\mathscr{C}}(G)$ the cardinality $\max_{C\in\mathscr{C}(G)}|C|$ of the largest clique among the cliques in $\mathscr{C}(G)$. For given vertex weights $0 \leqslant c_v \leqslant 1$, $v \in V$, we define $\omega_{\mathscr{C}}(G,c) = \max_{C\in\mathscr{C}(G)}\sum_{v\in C} c_v$. Note that $\omega_{\mathscr{C}}(G,c) \leqslant \omega(G,c)$.

Often the graph G will be given implicitly in another representation (e.g., G is taken to be the conflict graph of given paths in a network or of given intervals on the real line). In this case, we allow the relevant cliques $\mathscr{C}(G)$ to be defined in terms of the given representation of G; this is important, because in some applications it may be $\mathscr{NP}$-hard to construct a representation if only the graph is given (for example, this is the case for edge-intersection graphs of undirected paths in a tree [15]).

The *duplication* of a vertex v in an undirected graph $G = (V, E)$ is the operation of adding a new vertex v' to G and making v' adjacent to v and to all neighbors of v; if G is a weighted graph, the weight of v' is set equal to that of v. If a graph $G' = (V', E')$ is obtained from $G = (V, E)$ by deleting vertices and duplicating every remaining vertex an arbitrary number of times, we require sometimes that the set $\mathscr{C}(G')$ of relevant cliques in G' must contain exactly those cliques C' that can be obtained from a clique $C \in \mathscr{C}(G)$ by deleting and duplicating vertices in the same way as for obtaining G' from G. This requirement, which we call the *clique propagation* property, may seem restrictive, but it is satisfied automatically by natural definitions of $\mathscr{C}(G)$ in many applications (cf. Sections 4 and 5).

MIS and MWIS have already been defined at the beginning of the introduction. We assume that an instance of MWIS is given by an undirected graph $G = (V, E)$ with vertex weights, where the weight of vertex $v \in V$ is denoted by w_v. For $S \subseteq V$, we write $w(S)$ for $\sum_{v\in S} w_v$. A ρ-approximation algorithm for MWIS is an algorithm that runs in polynomial time and that always outputs an independent set whose weight is at least a $1/\rho$-fraction of the weight of an optimal solution. We will also encounter vertex weights in instances of coloring problems; in that case, the weight of vertex $v \in V$ will be denoted by c_v.

We consider the following linear program to compute a maximum weight fractional independent set subject to clique constraints for all cliques in $\mathscr{C}(G)$, using a variable x_v for every vertex $v \in V$ to indicate the fraction of vertex v that should be selected:

$$
\begin{array}{lll}
\mathrm{LP}_I & \max & \sum\limits_{v\in V} w_v x_v \\
 & \text{s.t.} & \sum\limits_{v\in C} x_v \leqslant 1 \quad \forall C \in \mathscr{C}(G), \\
 & & 0 \leqslant x_v \leqslant 1 \quad \forall v \in V.
\end{array}
$$

If $\mathscr{C}(G)$ is the set of all maximal cliques in G, the feasible solutions to LP_I form the so-called *fractional node-packing polytope QSTAB*(G) [17].

Given a graph $G = (V, E)$, a *coloring* of G is an assignment of colors to vertices such that adjacent vertices receive different colors. Note that every color class (set of vertices assigned the same color) is an independent set in G. The minimum number of colors required in a coloring is the *chromatic number* $\chi(G)$. It is clear that $\chi(G) \geqslant \omega(G)$. For a given graph $G = (V, E)$ with vertex weights $0 \leqslant c_v \leqslant 1$, $v \in V$, a *weighted fractional coloring* of G is a solution to the following linear program LP_w, where $\mathscr{J}$ is the set of all independent sets in G and a variable x_I indicates the "length" of the independent set I:

$$\begin{array}{lll}\mathrm{LP}_w & \min & \sum_{I \in \mathscr{J}} x_I \\ & \text{s.t.} & \sum_{I \in \mathscr{J} \mid v \in I} x_I \geqslant c_v \quad \forall v \in V, \\ & & 1 \geqslant x_I \geqslant 0 \quad \forall I \in \mathscr{J}.\end{array}$$

The goal is to compute independent sets I with fractions $x_I \in [0, 1]$ such that each vertex v is covered at least by c_v and such that the total length $\sum_{I \in \mathscr{J}} x_I$ is minimized. This minimum total length is the *weighted fractional chromatic number* $\chi_{\mathrm{f}}(G, c)$ of G with weight vector c. It is clear that $\omega(G, c)$ is a lower bound for $\chi_{\mathrm{f}}(G, c)$. Any feasible solution to LP_w is a weighted fractional coloring, and we refer to the corresponding objective value of LP_w as the *total length* of the coloring.

Note that the size of LP_w can be exponential in the size of the graph; an algorithm that computes a solution to LP_w in polynomial time (polynomial in the size of the graph and in the size of the representation of the vertex weights c_v) cannot construct LP_w explicitly.

If $c_v = 1$ for all $v \in V$, weighted fractional colorings are simply called *fractional colorings*. The optimal total length of a fractional coloring is the *fractional chromatic number* $\chi_{\mathrm{f}}(G)$ of G. Note that $\omega(G) \leqslant \chi_{\mathrm{f}}(G) \leqslant \chi(G)$.

We will use the following alternative interpretation of weighted fractional colorings: a weighted fractional coloring using independent sets of total length K can be seen as an assignment of left-open and right-closed subintervals of $(0, K]$ to vertices of G such that every vertex v is assigned disjoint subintervals of total length at least c_v and such that adjacent vertices are assigned disjoint subintervals.

2.1. Definition of tractable classes of graphs

Let $\mathscr{G}$ be a class of graphs that is closed under duplication and deletion of vertices. Let $\mathscr{C}$ be a function that maps every graph $G \in \mathscr{G}$ to a set of relevant cliques in G. For given $\mathscr{G}$ and $\mathscr{C}$, we say that LP_I can be solved in polynomial time with ratio $\beta \geqslant 1$ if there is an algorithm that, given a graph $G = (V, E)$ from $\mathscr{G}$ with weight vector w as input, computes a feasible solution to LP_I in polynomial time such that the objective value of the solution is at least a $1/\beta$-fraction of the optimal (fractional) solution. If the number of relevant cliques of a graph is polynomial in the size of the graph and if they can be determined in polynomial time, we can always solve LP_I optimally in polynomial time (so that $\beta = 1$), because LP_I is a linear program whose size is polynomial in the size of G and in the size of the representation of w in this case.

Definition 1. Let $\mathscr{G}$ be a class of graphs that is closed under duplication and deletion of vertices and let a set $\mathscr{C}(G)$ of relevant cliques be defined for every $G \in \mathscr{G}$. Assume that LP_I can be solved in polynomial time with ratio $\beta \geqslant 1$. Then we say that

(1) $(\mathscr{G}, \mathscr{C})$ is an (α, β)-*tractable* class of graphs if the clique propagation property holds and if there is a polynomial-time algorithm to solve the coloring problem using at most $\alpha\omega_{\mathscr{C}}(G)$ colors for every graph $G \in \mathscr{G}$,

(2) $(\mathscr{G},\mathscr{C})$ is (α,β)-*fractional-tractable* if the clique propagation property holds and if a fractional coloring of total length at most $\alpha\omega_{\mathscr{C}}(G)$ can be computed in polynomial time for every graph $G\in\mathscr{G}$, and

(3) $(\mathscr{G},\mathscr{C})$ is (α,β)-*weighted-fractional-tractable* if a weighted fractional coloring of total length at most $\alpha\omega_{\mathscr{C}}(G,c)$ can be computed in polynomial time for all graphs $G\in\mathscr{G}$ with arbitrary weight vectors c satisfying $0\leqslant c_v\leqslant 1$ for all $v\in V$.

Note that the clique propagation property is not required to hold for (α,β)-weighted-fractional-tractable classes of graphs.

If $\mathscr{C}(G)$ contains all maximal cliques in G, then $\omega_{\mathscr{C}}(G)=\omega(G)$. In this case, it is enough to assume in Definition 1 to have a coloring algorithm that computes a (weighted fractional) coloring with at most $\alpha\omega(G)$ colors (or of total length at most $\alpha\omega(G,c)$) for every graph $G\in\mathscr{G}$.

3. The conversion technique

In this section, we show that there are good approximation algorithms for MWIS in the classes of graphs defined in Definition 1.

Theorem 2. *For every (α,β)-tractable or (α,β)-fractional-tractable class of graphs, there is an approximation algorithm for MWIS that achieves approximation ratio $\alpha\beta+\varepsilon$, where ε is an arbitrarily small positive constant.*

Proof. As an (α,β)-tractable class of graphs is also (α,β)-fractional-tractable, it is enough to prove the theorem for (α,β)-fractional-tractable classes of graphs.

Let $(\mathscr{G},\mathscr{C})$ be an (α,β)-fractional-tractable class of graphs. Let $G=(V,E)$ be a graph from $\mathscr{G}$ with positive weights w_v for vertices $v\in V$. (We can assume without loss of generality that all weights are positive; vertices with non-positive weights can be deleted from G, because including such a vertex in an independent set would not increase the objective value of a solution.) Denote the optimum value of LP_I for this graph by z^*. By assumption, we have an algorithm B that computes a solution x^B of value $z^B\geqslant(1/\beta)z^*$ in polynomial time. Let $\varepsilon>0$ be an arbitrary positive constant. We define $\delta=\varepsilon/(\alpha\beta+\varepsilon)$ and $N=\lceil 4|V|/\delta^2\rceil$. Let $\bar{x}_v=(1-\delta/2)x_v^B+\delta/2|V|$. It is easy to see that $\bar{x}$ is a feasible solution of LP_I with objective value $\sum_{v\in V}w_v\bar{x}_v\geqslant(1-\delta/2)(1/\beta)z^*$. To get a solution $\hat{x}$, we round the $\bar{x}_v$ down to the largest possible multiple of $1/N$ (i.e., $\hat{x}_v=\lfloor\bar{x}_vN\rfloor/N$). We obtain that

$$\hat{x}_v\geqslant\bar{x}_v-\frac{1}{N}\geqslant\bar{x}_v-\frac{\delta^2}{4|V|}=\bar{x}_v-\frac{\delta}{2|V|}\frac{1}{2}\delta\geqslant\bar{x}_v(1-\delta/2),$$

using $\bar{x}_v\geqslant\delta/2|V|$. Since $\hat{x}$ is generated by rounding the $\bar{x}_v$ down to a multiple of $1/N$, feasibility of $\bar{x}$ for LP_I implies also feasibility of $\hat{x}$ for LP_I. The objective value $\hat{z}$ of $\hat{x}$ is given by $\sum_{v\in V}w_v\hat{x}_v\geqslant(1-\delta/2)\sum_{v\in V}w_v\bar{x}_v\geqslant(1-\delta/2)^2(1/\beta)z^*\geqslant(1-\delta)(1/\beta)z^*$.

So we have a rounded solution $\hat{x}$ to LP_I such that every $\hat{x}_v$ is of the form k/N for some $k\in\{0,1,\ldots,N\}$ and such that the objective value $\hat{z}\geqslant(1-\delta)(1/\beta)z^*$. Next, we construct a graph $\hat{G}$ by using the rounded solution $\hat{x}$ of LP_I. The set $\hat{V}$ of vertices of $\hat{G}$ contains, for every vertex $v \in V$, exactly k copies of v provided that $\hat{x}_v = k/N$. To obtain $\hat{G}$ from G, we replace vertex v by a clique A_v with k vertices such that each vertex in A_v has the same neighborhood (outside A_v) as v; in other words, vertex v is duplicated $k-1$ times if $k \geqslant 1$ and is deleted if $k = 0$.

By the clique propagation property, each relevant clique $\hat{C} \in \mathscr{C}(\hat{G})$ is generated from a clique $C \in \mathscr{C}(G)$, and this implies

$$|\hat{C}| = \sum_{v \in C} |A_v| = \sum_{v \in C} \hat{x}_v N \leqslant N.$$

Therefore, $\omega_{\mathscr{C}}(\hat{G}) \leqslant N$.

By assumption, we have an algorithm A_{f} that computes a fractional coloring of $\hat{G}$ with total length at most $\alpha\omega_{\mathscr{C}}(\hat{G})$ (and with only a polynomial number of independent sets $U_1, \ldots, U_y$). Since every vertex is covered by at least one, we have $\sum_{i=1}^{y} x_{U_i} w(U_i) \geqslant \hat{z}N$. Furthermore, the length $\sum_{i=1}^{y} x_{U_i} \leqslant \alpha N$. Suppose that all independent sets have weight $w(U_i) < \hat{z}/\alpha$. Then $\sum_{i=1}^{y} x_{U_i} w(U_i) < \sum_{i=1}^{y} x_{U_i} \hat{z}/\alpha = (\hat{z}/\alpha) \sum_{i=1}^{y} x_{U_i} \leqslant \hat{z}N$. This gives a contradiction. Therefore, there is at least one independent set U_i with $w(U_i) \geqslant \hat{z}/\alpha \geqslant [1/(\alpha\beta + \varepsilon)]z^*$. So we obtain an $(\alpha\beta + \varepsilon)$-approximation algorithm for MWIS by outputting the independent set with maximum weight among all independent sets used in the coloring. □

If the maximum weight fractional independent set problem LP_I is solvable in polynomial time, Theorem 2 gives an $(\alpha + \varepsilon)$-approximation algorithm for MWIS.

Theorem 3. *For every (α, β)-weighted-fractional-tractable class of graphs, there is an approximation algorithm for MWIS that achieves approximation ratio $\alpha\beta$.*

Proof. Let $(\mathscr{G}, \mathscr{C})$ be an (α, β)-weighted-fractional-tractable class of graphs. By assumption, we have an algorithm B that computes a solution x^B to LP_I of value $z^B \geqslant (1/\beta)z^*$ in polynomial time. Furthermore, we have an algorithm A_{f} to compute weighted fractional colorings with total length at most $\alpha\omega_{\mathscr{C}}(G, c)$ for any vertex weight vector c.

After obtaining solution x^B to LP_I, we set $c_v = x_v^B$ for all $v \in V$ and use A_{f} to compute a weighted fractional coloring of (G, x^B) with total length at most $\alpha\omega_{\mathscr{C}}(G, x^B)$ and with a polynomial number of independent sets $U_1, \ldots, U_y$. This solution of LP_w satisfies $\sum_{i=1}^{y} w(U_i) x_{U_i} \geqslant z^B$. Furthermore, we can bound

$$\omega_{\mathscr{C}}(G, x^B) = \max_{C \in \mathscr{C}(G)} \sum_{v \in C} x_v^B \leqslant 1,$$

since x^B is a feasible solution of LP_I. Now suppose that all independent sets in the solution of algorithm A_{f} have weight less than z^B/α. This implies that

$$z^B \leqslant \sum_{i=1}^{y} w(U_i) x_{U_i} < \frac{z^B}{\alpha} \sum_{i=1}^{y} x_{U_i} \leqslant z^B \omega_{\mathscr{C}}(G, x^B) \leqslant z^B$$

and gives a contradiction. Therefore, one of the independent sets has weight $\geqslant z^B/\alpha \geqslant z^*/\alpha\beta$. By outputting the independent set of maximum weight among all independent sets used in the coloring, we obtain an $\alpha\beta$-approximation algorithm for MWIS. □

Again, if the fractional weighted independent set problem is solvable in polynomial time, we obtain an α-approximation algorithm for MWIS.

In the remaining sections, we discuss applications of Theorems 2 and 3.

4. Weighted edge-disjoint paths problems

Modern high-performance communication networks usually support bandwidth reservation: when a connection is established, the required bandwidth is reserved on all links along a path from the sender to the receiver. For a given set of connection requests, *call control* is the problem of deciding which of the requests should be accepted and which should be rejected. An important special case of call control arises if all connection requests have bandwidth requirements that are larger than half the link capacity. In this case, no two connections can use the same link at the same time, so simultaneously active connections must be routed along edge-disjoint paths. We study the case where each path is associated with a certain weight (value) and the goal is to maximize the total weight of the accepted paths.

A directed graph $G=(V,E)$ is *bidirected* if $(u,v)\in E$ implies $(v,u)\in E$. A bidirected tree is the graph obtained from an undirected tree by replacing each edge by two directed edges with opposite directions. A bidirected mesh is obtained in the same way from an undirected mesh. Here (and in the next section) we assume that all paths in meshes are row–column paths (dimension-order paths), i.e., paths that consist of a horizontal segment in the row of the starting node followed by a vertical segment in the column of the end node of the path. Row–column routing is often used in practice because it is deadlock-free and can be implemented with simple hardware.

An instance of the MWEDP consists of a bidirected graph $G=(V,E)$ and a set P of directed paths in G, where each path $p\in P$ has a weight $w_p>0$. A feasible solution is a subset of the given paths (the *accepted* paths) such that the paths in that subset are edge-disjoint.

Given a set P of directed paths in a bidirected graph $G=(V,E)$, the *load* $L(e)$ of edge $e\in E$ is the number of paths in P that go through edge e. The *maximum load* L is then defined as $\max_{e\in E} L(e)$. A *coloring* of the paths in P is an assignment of colors such that paths receive different colors if they share a directed edge. The *path coloring* problem is the problem of coloring a given set of paths with the minimum number of colors. Obviously, the maximum load is a lower bound on the number of colors in an optimal coloring.

4.1. Previous work

The unweighted version of MWEDP (where all paths have the same weight), called MEDP, has been studied for tree networks in [9]. It was shown that MEDP can be solved optimally in polynomial time for bidirected trees of constant degree and is MAX SNP-hard for bidirected trees of arbitrary degree. For every fixed constant $\varepsilon>0$, a $(\frac{5}{3}+\varepsilon)$-approximation algorithm was presented.

Interestingly, MWEDP can be solved optimally in polynomial time in undirected trees (the given paths are undirected paths in this case as well) of arbitrary degree [33].

As far as we know, MEDP and MWEDP have not been studied in meshes with row–column routing before. It is easy to see that for arbitrary (but fixed as part of the input) paths in mesh networks, MEDP is as hard to approximate as the maximum independent set problem on general graphs. This suggests to study the problem for restricted types of paths, such as row–column paths.

MEDP has also been studied in the case where each connection request specifies only the sender and the receiver, and the path can be selected by the algorithm. For arbitrary directed graphs with m edges, this problem was recently shown to be $\mathcal{NP}$-hard to approximate within $m^{1/2-\varepsilon}$ [18]. Approximation algorithms with approximation ratio $\mathrm{O}(\sqrt{m})$ are known for the unweighted case [24,32] and for the weighted case [26]. Better approximation ratios can be achieved for restricted classes of graphs. For a class of planar graphs containing two-dimensional mesh networks, an $\mathrm{O}(1)$ approximation algorithm has been devised in [25].

The path coloring problem has been studied as well. The best known approximation algorithm for coloring directed paths in bidirected trees uses at most $\lceil 5L/3 \rceil$ colors, where L is the maximum load [11].

4.2. Approximating MWEDP in trees and meshes

We give a $(\frac{5}{3}+\varepsilon)$-approximation for MWEDP in trees (matching the performance of the approximation algorithm for the unweighted case in [9]) and a 4-approximation for MWEDP in meshes with row–column routing.

Let $\mathcal{G}$ be the class of all conflict graphs obtained from sets of directed paths in bidirected trees. If G is the conflict graph (edge-intersection graph) of a given set of directed paths in a bidirected tree, we let $\mathscr{C}(G)$ be the set of all cliques in G corresponding to paths using the same directed edge of the tree. This means that $\mathscr{C}(G)$ contains one clique for every directed edge of the tree. Note that the clique propagation property is satisfied. Obviously, $\omega_{\mathscr{C}}(G) = L$, where L is the maximum load. LP_I has polynomial size and can be solved optimally in polynomial time ($\beta = 1$), and the coloring algorithm from [11] uses at most $5L/3$ colors ($\alpha = \frac{5}{3}$). Therefore, $(\mathcal{G}, \mathscr{C})$ is a $(\frac{5}{3}, 1)$-tractable class of graphs, and by Theorem 2 we get a $(\frac{5}{3}+\varepsilon)$-approximation algorithm for MWEDP in bidirected trees.

Now we consider meshes with row–column routing. First, we can show that MEDP and path coloring are both $\mathcal{NP}$-hard for these networks:

Theorem 4. *MEDP is MAX SNP-hard for meshes with row–column routing. Path coloring is $\mathcal{NP}$-hard for meshes with row–column routing even if the number of rows or the number of columns is three*.

The proofs for the two claims of Theorem 4 are given in Appendix A and Appendix B, respectively.

Let $\mathcal{G}$ be the class of all conflict graphs obtained from sets of directed row–column paths in bidirected meshes. If G is the conflict graph (edge-intersection graph) of a given set of directed row–column paths in a bidirected mesh, we let $\mathscr{C}(G)$ be the set of all cliques in G corresponding to paths using the same directed edge of the mesh. LP_I can again be solved optimally in polynomial time ($\beta = 1$), and the following procedure computes a fractional weighted coloring using independent sets of total length at most $4\omega_{\mathscr{C}}(G, x^B)$: partition the given paths into the sets right-up, right-down, left-up, and left-down. Vertical and horizontal paths can be put in any of the two possible sets. The paths in each of the four sets can be colored using independent sets of total length at most $2\omega_{\mathscr{C}}(G, x^B)$, as will be shown below. Then we can use the same colors for the sets right-up and left-down, and for the sets right-down and left-up. Thus, a weighted fractional coloring of total length at most $4\omega_{\mathscr{C}}(G, x^B)$ is obtained.

We explain how to color the paths in one set, say in the set right-up. Consider the nodes of the mesh starting from the bottom right corner, processing the nodes in one row from right to left and processing the rows from bottom to top. At each node, consider the paths that turn (or start or end) at that node in arbitrary order, and assign each of them free subintervals of free colors greedily. Let p denote the currently processed path. In order to color p, we must assign disjoint subintervals of $(0, 2\omega_{\mathscr{C}}(G, x^B)]$ of total length x_p^B to p. As previously colored paths that intersect the current path must intersect its last horizontal edge or its first vertical edge, at most $2(\omega_{\mathscr{C}}(G, x^B) - x_p^B)$ of the available colors (i.e., of the interval $(0, 2\omega_{\mathscr{C}}(G, x^B)]$) are blocked, and free intervals of the required total length can always be found.

We note that the coloring procedure can be implemented in polynomial time. At any time during the coloring procedure, a certain number of interval endpoints has been used for specifying subintervals so far. When path p is colored, the required subintervals can be chosen such that at most one new interval endpoint is introduced; hence, there will be only $O(|P|)$ interval endpoints, and they can be processed in polynomial time.

So we have proved that $(\mathscr{G}, \mathscr{C})$ is $(4, 1)$-weighted-fractional-tractable, and by Theorem 3 we get a 4-approximation for MWEDP in meshes with row–column routing. Our approximation results for MWEDP are summarized in the following theorem.

Theorem 5. *There are approximation algorithms with approximation ratio* $\frac{5}{3} + \varepsilon$ (*for every fixed* $\varepsilon > 0$) *for MWEDP in bidirected trees and with approximation ratio* 4 *for MWEDP in bidirected meshes with row–column routing*.

5. Time-constrained scheduling of weighted packets

In networks with real-time requirements packets must be delivered before their individual deadlines. Applications are, for example, traffic control systems where data from sensors must be delivered to decision units in a timely manner. Assuming that each packet is associated with a certain weight (importance), the goal is to maximize the total weight of packets that are delivered by their deadlines.

For time-constrained packet scheduling, every packet p is specified by a directed path P_p, a release time r_p, a deadline d_p, and a weight $w_p > 0$. The number of links in P_p is denoted by $|P_p|$. We consider the bufferless case, where a packet must continue to travel one link in every time step after it has left the sender until it reaches the receiver. (In the buffered case, a packet can wait at intermediate nodes for an arbitrary number of time steps before it traverses the next link.) Transmitting a packet across one link takes exactly one time step, and every link can transmit at most one packet in every time step. A node can send and receive several packets simultaneously (multi-port model). An instance of TCSWP consists of a bidirected graph and a set of packets. Let n denote the number of packets. A feasible solution selects a subset of the given packets (the *accepted* packets) and specifies a starting time $s_p \geqslant r_p$ for every accepted packet, where every accepted packet must reach its destination by its deadline (i.e., $s_p + |P_p| \leqslant d_p$) and no two packets can use the same link in the same time step. The value of a solution is the sum of the weights of the accepted packets.

5.1. Previous work

The time-constrained packet scheduling problem was studied by Adler et al. for unweighted packets in linear networks (chains) [2]. They proved the problem $\mathcal{NP}$-hard in the bufferless case and presented a 2-approximation algorithm. In [1], Adler et al. studied the problem for weighted packets in trees and in meshes with row–column routing. For packets with arbitrary weights, they presented an 18-approximation algorithm for trees and an $\mathrm{O}(1)$-approximation algorithm for meshes with row–column routing. For the unweighted case, they gave a 3-approximation algorithm for trees and an $\mathrm{O}(1)$-approximation algorithm for meshes with row–column routing. The buffered case was also studied in [2] and [1]. It was proved that the value of an optimal solution is increased by at most a logarithmic factor (logarithmic in certain parameters of the network and the number of packets in an optimal solution) if buffering at intermediate nodes is allowed. Hence, approximation algorithms for the bufferless case are also approximation algorithms for the buffered case, only the ratio is worse by a logarithmic factor. It is an interesting open problem to find approximation algorithms for the buffered case that do not rely on this reduction to the bufferless case.

5.2. Approximating TCSWP in trees and meshes

Our technique allows us to obtain a 3-approximation for TCSWP in tree networks (improving on the previously known 18-approximation algorithm and matching the ratio of the approximation algorithm for the unweighted case [1]) and a 6-approximation for meshes with row–column routing (improving the $\mathrm{O}(1)$-approximation in [1]). We observe that TCSWP is MAX SNP-hard for tree networks and for meshes with row–column routing: the reductions that establish MAX SNP-hardness for MEDP in trees [9] and in meshes with row–column routing (Appendix A) can be modified for TCSWP by replacing paths by packets in a rather straightforward way.

Let t be a possible starting time for packet p, i.e., $r_p \leqslant t \leqslant d_p - |P_p|$. The *t-realization* of packet p is a packet that traverses the ith link of P_p in time step $t + i - 1$, $1 \leqslant i \leqslant |P_p|$. The t_1-realization of packet p_1 and the t_2-realization of packet p_2 are in conflict either if $p_1 = p_2$ or if there is a directed edge e that is used in the same time step by both realizations. Now, the real-time packet scheduling problem can be viewed as the problem of selecting a maximum weight subset of all possible realizations of all given packets such that no two selected realizations are in conflict.

Given a set Q of realizations of packets, a coloring of Q is an assignment of colors to elements of Q such that realizations that are in conflict receive different colors.

For a given instance of TCSWP, the number of possible packet realizations may be exponential in the size of the input. However, this is not a problem because of the following arguments: in trees and in meshes with row–column routing, a packet p with $d_p - |P_p| -$

$r_p > n$ can always be scheduled: after all $n-1$ other packets are scheduled, there still exists at least one starting time s_p for packet p such that the s_p-realization of p is not in conflict with any of the other scheduled packets. Therefore, all packets p with $d_p - |P_p| - r_p > n$ can be removed from the instance and treated separately. Then $d_p - r_p$ is polynomial in the size of the input for all remaining packets, and only a polynomial number of packet realizations must be considered.

First, let $\mathscr{G}$ be the class of all graphs obtained as conflict graphs of packet realizations in bidirected tree networks. For a graph $G \in \mathscr{G}$, let $\mathscr{C}(G)$ be the set of all cliques containing packet realizations using the same edge of the network in the same time step and all cliques containing packet realizations belonging to the same original packet. $(\mathscr{G}, \mathscr{C})$ is a $(3, 1)$-weighted-fractional-tractable class of graphs: LP_I has polynomial size and can be solved optimally in polynomial time, and the following procedure colors any weighted graph in $\mathscr{G}$ using independent sets of total length at most $3\omega_{\mathscr{C}}(G, x^B)$: process the nodes of the tree in a pre-order traversal starting at the root. At each node v, consider the packet realizations whose paths touch (start, end, or go through) the node v but no node that is closer to the root. Assign each such realization r the smallest available free color intervals. It is easy to see that, if a previously colored packet realization uses an edge in the same time step as r, it must in fact use an edge incident to v in the same time step as r. Hence, at most $3(\omega_{\mathscr{C}}(G, x^B) - x_r^B)$ of the available colors are used by previously colored realizations that are in conflict with r, namely at most $2(\omega_{\mathscr{C}}(G, x^B) - x_r^B)$ by realizations using an edge incident to v in the same time step as r and at most $\omega_{\mathscr{C}}(G, x^B) - x_r^B$ by realizations belonging to the same packet as r. Therefore, $3\omega_{\mathscr{C}}(G, x^B)$ colors suffice to complete the weighted fractional coloring. Hence, $(\mathscr{G}, \mathscr{C})$ is a $(3, 1)$-weighted-fractional-tractable class of graphs, and by Theorem 3 we obtain a 3-approximation algorithm for TCSWP in bidirected trees.

Now let $\mathscr{G}$ be the class of all graphs obtained as conflict graphs of packet realizations in bidirected mesh networks. For a graph $G \in \mathscr{G}$, let $\mathscr{C}(G)$ be defined as above. $(\mathscr{G}, \mathscr{C})$ is a $(6, 1)$-weighted-fractional-tractable class of graphs: LP_I can again be solved optimally in polynomial time, and a procedure for coloring the packet realizations similar to the method presented for paths in meshes (partitioning the packet realizations into the sets right-up, right-down, left-up, and left-down; and then coloring the packet realizations in each set greedily in the same order as the paths in Section 4) colors any weighted graph in $\mathscr{G}$ using independent sets of total length at most $6\omega_{\mathscr{C}}(G, x^B)$ (we get the factor 6 instead of 4, because we need $3\omega_{\mathscr{C}}(G, x^B)$ colors for each set instead of $2\omega_{\mathscr{C}}(G, x^B)$: the length of colors that are blocked for the current packet realization r by previously colored packet realizations is at most $3\omega_{\mathscr{C}}(G, x^B) - 3x_r^B$, namely $\omega_{\mathscr{C}}(G, x^B) - x_r^B$ by conflicting packet realizations on each of the two relevant edges and $\omega_{\mathscr{C}}(G, x^B) - x_r^B$ by packet realizations of the same packet). By Theorem 3 we get a 6-approximation algorithm for TCSWP in bidirected two-dimensional mesh networks if the paths are restricted to be row–column paths. The approximation results of this section are summarized in the following theorem.

Theorem 6. *There are approximation algorithms with approximation ratio* 3 *for TCSWP in bidirected trees and with approximation ratio* 6 *for TCSWP in bidirected meshes with row–column routing.*

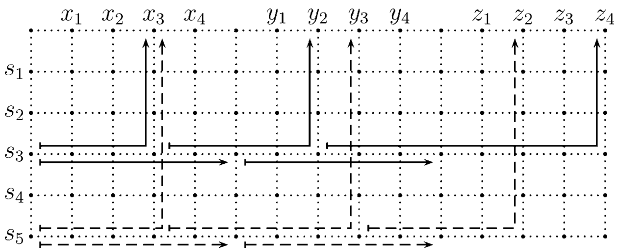

Fig. A.1. Reduction from MAXIMUM 3D-MATCHING (paths for $s_3 = (x_3, y_2, z_4)$ drawn solid, paths for $s_5 = (x_3, y_3, z_2)$ drawn dashed).

We note that our approximation algorithms for TCSWP work also if every packet does not specify a release time and a deadline, but an arbitrary (possibly non-contiguous) set of possible starting times.

Acknowledgements

The authors are grateful to Baruch Schieber and Seffi Naor for bringing the work in [5] to their attention and for helpful discussions.

Appendix A. MAX SNP-hardness of MEDP in meshes

In this appendix we prove the first claim of Theorem 4 (MAX SNP-hardness of MEDP in meshes with row–column routing) using a reduction from the MAXIMUM 3D-MATCHING problem. An instance of MAXIMUM 3D-MATCHING is given by three disjoint sets X, Y, Z with $|X| = |Y| = |Z|$ and a set of triples $S \subseteq X \times Y \times Z$. A feasible solution is a subset $M \subseteq S$ consisting of disjoint triples. The goal is to compute a feasible solution $M \subseteq S$ that maximizes $|M|$. It is known that MAXIMUM 3D-MATCHING is MAX SNP-complete even if each element of $X \cup Y \cup Z$ occurs in at most a constant number of triples in S [23].

Let an instance I of MAXIMUM 3D-MATCHING be given. Take $n = |X|$ and $m = |S|$. We construct an instance I' of MEDP in a bidirected mesh. The mesh has $m+1$ rows (numbered from 0 to m, where 0 is the top row) and $3n + 3$ columns (numbered from 0 to $3n + 2$, where 0 is the leftmost column). See Fig. A.1 for an example with $n = 4$ and $m = 5$. Let $X = \{x_1, \dots, x_n\}$, $Y = \{y_1, \dots, y_n\}$, $Z = \{z_1, \dots, z_n\}$, and $S = \{s_1, \dots, s_m\}$. For $1 \leqslant i \leqslant n$ (resp. $n+2 \leqslant i \leqslant 2n+1$ and $2n+3 \leqslant i \leqslant 3n+2$), column i of the mesh corresponds to the element x_i (resp. y_{i-n-1} and z_{i-2n-2}). For $1 \leqslant j \leqslant m$, row j of the mesh corresponds to the triple s_j. A node in column x and row y of the mesh is denoted by (x, y).

For each triple $s_j = (x_f, y_g, z_h) \in S$, we introduce five paths: a path from $(0, j)$ to $(f, 0)$, a path from (f, j) to $(n + 1 + g, 0)$, a path from $(n + 1 + g, j)$ to $(2n + 2 + h, 0)$, a path from $(0, j)$ to $(n + 1, j)$, and a path from $(n + 1, j)$ to $(2n + 2, j)$. Note that the first three paths for triple s_j are turning and that the remaining two are horizontal. The paths for two triples $s_3 = (x_3, y_2, z_4)$ and $s_5 = (x_3, y_3, z_2)$ are shown in Fig. A.1. So the set P of paths in instance I' contains $5m$ paths.

Observe that any feasible solution $Q \subseteq P$ to I' can contain at most 3 out of the 5 paths for each triple s_j. If Q contains three paths for a triple $s_j = (x_f, y_g, z_h)$, these must be the three turning paths, and there cannot be any other paths in Q using edges in columns f, $n+1+g$, and $2n+2+h$ of the mesh. On the other hand, if Q contains at most two paths for a triple s_j, we can assume without loss of generality that Q contains both horizontal paths for s_j, because these two paths do not interfere with any paths for the other triples.

We claim that a feasible solution to I consisting of t triples can be converted into a feasible solution to I' consisting of $2m+t$ paths, and vice versa. Given a feasible solution to I with t triples, we obtain the solution to I' by taking the three turning paths for each of the t triples and the two horizontal paths for each remaining triple. Given a feasible solution Q to I' with $2m+t$ paths, we obtain the solution to I by selecting all triples for which three paths are contained in Q.

This already proves that MEDP is $\mathcal{NP}$-hard for meshes with row–column routing. To establish MAX SNP-hardness, consider the variant of MAXIMUM 3D-MATCHING where the number of occurrences of an element of $X \cup Y \cup Z$ in triples of S is bounded by a constant k. Let t^* denote the number of triples in an optimal solution to I. Then $t^* \geqslant m/(3k-2)$, because the greedy algorithm will produce at least this many triples. Therefore, an optimal solution to I' is larger than an optimal solution to I by at most a constant factor (note that $2m + t^* \leqslant 2(3k-2)t^* + t^* \leqslant (6k-3)t^*$), and an approximate solution to I' that differs from an optimal solution to I' by an absolute value of x yields an approximate solution to I that differs from an optimal solution to I by at most x. This shows that the reduction explained above is an L-reduction (see [30]). Thus we obtain that MEDP is MAX SNP-hard for meshes with row–column routing.

Note that our proof uses only paths that go right and paths that go right and then up. Hence, MEDP in meshes is already MAX SNP-hard if only paths of this type are allowed. Furthermore, we remark that MWEDP can be solved optimally in polynomial time if the number of rows (or the number of columns) of the mesh is bounded by a constant. In that case, the mesh is a graph with bounded treewidth and with bounded degree, and MWEDP is polynomial on such graphs (as remarked in [8, p. 201]).

Appendix B. NP-hardness of path coloring in meshes

In this appendix, we prove the second claim of Theorem 4, i.e., we prove that path coloring is $\mathcal{NP}$-hard in meshes with row–column routing even if the number of rows or the number of columns is three.

We use a reduction from the $\mathcal{NP}$-complete ARC COLORING problem [14]. A graph $G = (V, E)$ is a circular arc graph [34] if its vertices can be represented by arcs of a circle such that there is an edge between two vertices in G if and only if the corresponding arcs intersect. An instance I of ARC COLORING is given by a family $F = \{A_1, A_2, \ldots, A_n\}$ of circular arcs and a positive integer K. The problem is to decide whether F can be colored with K colors such that intersecting arcs are assigned different colors.

Each arc $A_i \in F$ is given by a pair $\langle a_i, b_i \rangle$ with $a_i, b_i \in \{0, \ldots, m\}$ and $a_i \neq b_i$. Intuitively, the set $\{0, \ldots, m\}$ represents points that are located consecutively around a circle. The *span* $sp(A_i)$ of arc A_i is the set $\{a_i, a_i + 1, \ldots, b_i\}$ if $a_i < b_i$ and $\{a_i, a_i +$

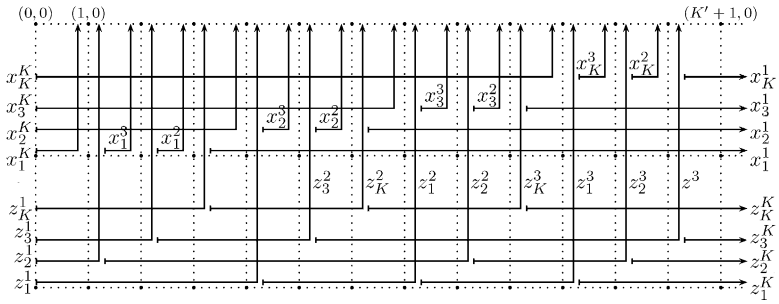

Fig. B.1. Reduction from ARC COLORING: the left part of the mesh.

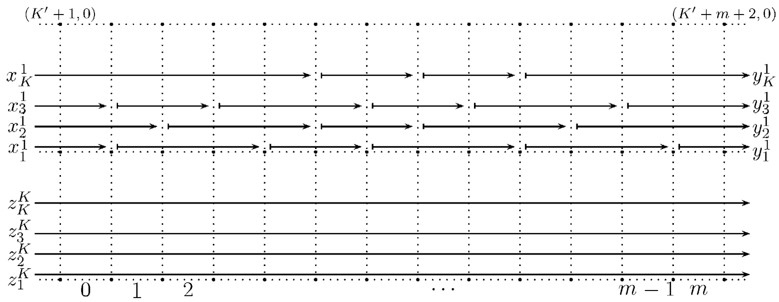

Fig. B.2. Reduction from ARC COLORING: the middle part of the mesh.

$1, \ldots, m, 0, \ldots, b_i\}$ if $a_i > b_i$. Two arcs A_i and A_j intersect if $sp(A_i) \cap sp(A_j) \neq \emptyset$. Without loss of generality, we can assume that $a_i > b_i$ for exactly the first K arcs $A_1, \ldots, A_K$.

We transform an instance I of ARC COLORING into an instance I' of the path coloring problem that can be colored with K colors if and only if I can be colored with K colors. Let $K' = K(K-1)$. Consider a mesh with 3 rows (numbered from 0 to 2) and $2K' + m + 4$ columns. See Figs. B.1–B.3 for an example with $K = 4$. Intuitively, we cut the arcs of instance I between points m and 0 and represent the resulting intervals by paths in row 1 in the middle part of the mesh. Then we use a number of auxiliary paths to ensure that the two paths corresponding to the same original arc receive the same color in any coloring with K colors.

More precisely, we proceed as follows. For the first K arcs $A_i \in F$ we define two paths x_i^1 and y_i^1 such that:

(1) x_i^1 comes from node $(K', 1)$ (the exact starting node will be determined below) and terminates at node $(K' + 2 + b_i, 1)$,

(2) y_i^1 starts at node $(K' + 1 + a_i, 1)$ and leads to node $(K' + m + 2, 1)$ (the final destination of the path will be determined below).

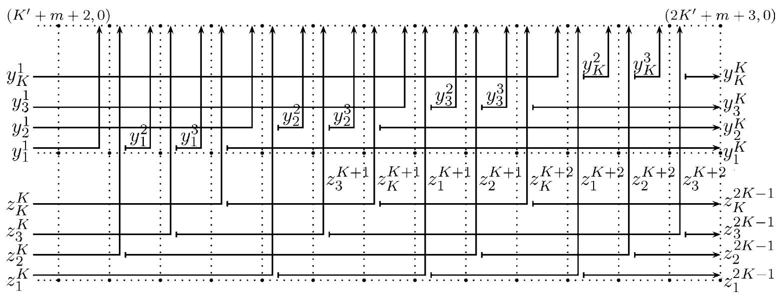

Fig. B.3. Reduction from ARC COLORING: the right part of the mesh.

For the other arcs $A_i \in F$ with $a_i < b_i$ we use one path c_i from node $(K' + 1 + a_i, 1)$ to node $(K' + 2 + b_i, 1)$ (see Fig. B.2). Note that the conflict graph corresponding to the set of paths $\{x_i^1, y_i^1 \mid 1 \leqslant i \leqslant K\} \cup \{c_i \mid K + 1 \leqslant i \leqslant n\}$ is an interval graph (that can be colored with K colors).

Next, we extend the paths x_i^1 and y_i^1 and introduce further paths z_i^j (for $1 \leqslant i \leqslant K$, $1 \leqslant j \leqslant 2K - 1$), x_i^j, y_i^j (for $1 \leqslant i \leqslant K$, $1 < j \leqslant K$) as shown in Figs. B.1 and B.3. We have constructed an instance I' with 3 rows and $2K(K - 1) + m + 4$ columns. Since $K \leqslant n$ and $m = \mathrm{O}(n)$, we get a mesh with a polynomial number of rows and columns and a polynomial number of paths. Therefore, instance I' can be constructed in polynomial time.

Suppose that there exists a K-coloring f for the instance I'. Clearly, we must have

$$\{f(z_1^1), f(z_2^1), \ldots, f(z_K^1)\} = \{1, \ldots, K\}.$$

Without loss of generality we may assume that $f(z_i^1) = i$ for $1 \leqslant i \leqslant K$. Using the conflict structure of the paths in row 2 of the mesh we obtain $f(z_i^j) = i$ for each $1 \leqslant j \leqslant 2K - 1$ and each $1 \leqslant i \leqslant K$. For example, consider the second path z_2^2 (Fig. B.1). This path is in conflict with $z_1^1, z_3^1, \ldots, z_K^1$ and, therefore, must be colored with color 2 (the same color as $f(z_2^1)$). This means that the colors $1, \ldots, K$ are propagated in row 2.

Using the same idea, we have $f(y_i^1) = f(y_i^2) = \cdots = f(y_i^K)$ for each $1 \leqslant i \leqslant K$. Since the y_i-paths are in conflict with paths of colors $1, \ldots, i - 1, i + 1, \ldots, K$, it follows that $f(y_i^1) = i$. The same idea shows also that $f(x_i^1) = i$ for $1 \leqslant i \leqslant K$. Since $f(x_i^1) = f(y_i^1) = i$ for $1 \leqslant i \leqslant K$, the coloring restricted to the set of paths $\{x_i^1, y_i^1 \mid 1 \leqslant i \leqslant K\} \cup \{c_i \mid K + 1 \leqslant i \leqslant n\}$ gives a K-coloring of the instance I (the circular arc graph).

Conversely, a K-coloring for I can be extended directly to a K-coloring for I'. Since ARC COLORING is $\mathcal{NP}$-hard and the reduction can be done in polynomial time, the path coloring problem in meshes with row–column routing is $\mathcal{NP}$-hard even for meshes with three rows. Furthermore, the instance I' can be converted into an equivalent instance of path coloring in a mesh with three columns by reversing all paths and rotating the mesh by 90°. Therefore, path coloring in meshes with row–column routing is also $\mathcal{NP}$-hard for meshes with three columns. This completes the proof.

References

[1] M. Adler, S. Khanna, R. Rajaraman, A. Rosén, Time-constrained scheduling of weighted packets on trees and meshes, in: Proceedings of the 11th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA'99), 1999, pp. 1–12.

[2] M. Adler, A.L. Rosenberg, R.K. Sitaraman, W. Unger, Scheduling time-constrained communication in linear networks, in: Proceedings of the 10th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA'98), 1998, pp. 269–278.

[3] S. Arnborg, A. Proskurowski, Linear time algorithms for NP-hard problems restricted to partial k-trees, Discrete Appl. Math. 23 (1989) 11–24.

[4] A. Bar-Noy, R. Bar-Yehuda, A. Freund, J.S. Naor, B. Schieber, A unified approach to approximating resource allocation and scheduling, J. ACM 48 (5) (2001) 1069–1090.

[5] A. Bar-Noy, S. Guha, J.S. Naor, B. Schieber, Approximating the throughput of multiple machines in real-time scheduling, SIAM J. Comput. 31 (2) (2001) 331–352.

[6] P. Berman, B. DasGupta, Multi-phase algorithms for throughput maximization for real-time scheduling, J. Combin. Optim. 4 (3) (2000) 307–323.

[7] P. Berman, T. Fujito, On the approximation properties of independent set problem in degree 3 graphs, in: Proceedings of the Fourth International Workshop on Algorithms and Data Structures (WADS'95), Lecture Notes in Computer Science, vol. 955, Springer, Berlin, 1995, pp. 449–460.

[8] T. Erlebach, Scheduling connections in fast networks, Ph.D. Thesis, Technische Universität München, 1999.

[9] T. Erlebach, K. Jansen, The maximum edge-disjoint paths problem in bidirected trees, SIAM J. Discrete Math. 14 (3) (2001) 326–355.

[10] T. Erlebach, K. Jansen, Implementation of approximation algorithms for weighted and unweighted edge-disjoint paths in bidirected trees, ACM J. Exp. Algorithmics 7 (2002).

[11] T. Erlebach, K. Jansen, C. Kaklamanis, M. Mihail, P. Persiano, Optimal wavelength routing on directed fiber trees (special issue of ICALP'97), Theoret. Comput. Sci. 221 (1999) 119–137.

[12] U. Feige, J. Kilian, Zero knowledge and the chromatic number, J. Comput. System Sci. 57 (2) (1998) 187–199.

[13] M.R. Garey, D.S. Johnson, Computers and Intractability. A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, New York, San Francisco, 1979.

[14] M.R. Garey, D.S. Johnson, G.L. Miller, C.H. Papadimitriou, The complexity of coloring circular arcs and chords, SIAM J. Algebraic Discrete Methods 1 (2) (1980) 216–227.

[15] M.C. Golumbic, R.E. Jamison, Edge and vertex intersection of paths in a tree, Discrete Math. 55 (1985) 151–159.

[16] M. Grötschel, L. Lovász, A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, Combinatorica 1 (1981) 169–197 (corrigendum in Combinatorica 4, 291–295).

[17] M. Grötschel, L. Lovász, A. Schrijver, Geometric Algorithms and Combinatorial Optimization, Springer, Berlin, 1988.

[18] V. Guruswami, S. Khanna, R. Rajaraman, B. Shepherd, M. Yannakakis, Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems, in: Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC'99), 1999, pp. 19–28.

[19] M.M. Halldórsson, Approximations of independent sets in graphs, in: Proceedings of the First International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX'98), Lecture Notes in Computer Science, vol. 1444, Springer, Berlin, 1998, pp. 1–13.

[20] M.M. Halldórsson, Approximations of weighted independent set and hereditary subset problems, in: Proceedings of the Fifth International Computing and Combinatorics Conference (COCOON'99), Lecture Notes in Computer Science, vol. 1627, Springer, Berlin, 1999, pp. 261–270.

[21] J. Håstad, Clique is hard to approximate within $n^{1-\varepsilon}$, Acta Math. 182 (1999) 105–142.

[22] D. Kagaris, S. Tragoudas, Maximum weighted independent sets on transitive graphs and applications, Integration 27 (1999) 77–86.

[23] V. Kann, Maximum bounded 3-dimensional matching is MAX SNP-complete, Inform. Process. Lett. 37 (1991) 27–35.

[24] J. Kleinberg, Approximation algorithms for disjoint paths problems, Ph.D. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1996.

[25] J. Kleinberg, É. Tardos, Disjoint paths in densely embedded graphs, in: Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS'95), 1995, pp. 52–61.
[26] S.G. Kolliopoulos, C. Stein, Approximating disjoint-path problems using greedy algorithms and packing integer programs, in: Proceedings of the Sixth Integer Programming and Combinatorial Optimization Conference (IPCO VI), Lecture Notes in Computer Science, vol. 1412, Springer, Berlin, 1998, pp. 153–168.
[27] L. Lovász, On the ratio of the optimal integral and fractional covers, Discrete Math. 13 (1975) 383–390.
[28] C. Lund, M. Yannakakis, On the hardness of approximating minimization problems, J. ACM 41 (5) (1994) 960–981.
[29] E. Malesińska, Graph-theoretical models for frequency assignment problems, Ph.D. Thesis, Technische Universität Berlin, 1997.
[30] C. Papadimitriou, M. Yannakakis, Optimization, approximization and complexity classes, J. Comput. System Sci. 43 (1991) 425–440.
[31] F. Spieksma, Approximating an interval scheduling problem, in: Proceedings of the First International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX'98), Lecture Notes in Computer Science, vol. 1444, Springer, Berlin, 1998, pp. 169–180.
[32] A. Srinivasan, Improved approximations for edge-disjoint paths, unsplittable flow, and related routing problems, in: Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS'97), 1997, pp. 416–425.
[33] R.E. Tarjan, Decomposition by clique separators, Discrete Math. 55 (1985) 221–232.
[34] A. Tucker, Coloring a family of circular arcs, SIAM J. Appl. Math. 29 (3) (1975) 493–502.
[35] B. Verweij, K. Aardal, An optimisation algorithm for maximum independent set with applications in map labelling, in: Proceedings of the Seventh Annual European Symposium on Algorithms (ESA'99), Lecture Notes in Computer Science, vol. 1643, Springer, Berlin, 1999, pp. 426–437.